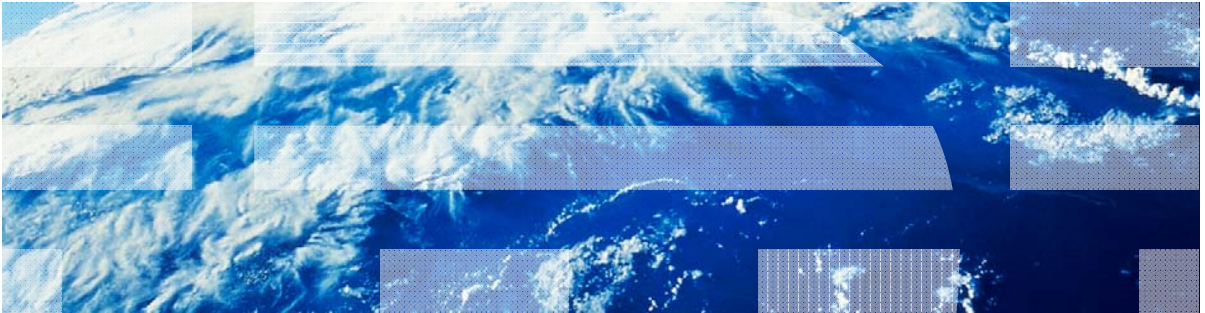IBM

IBM Worklight V6.0.0
Getting Started

Integrating server-generated pages in hybrid applications

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

- Other company products or service names may be trademarks or service marks of others.

- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See http://www.ibm.com/ibm/us/en/

Agenda

- Overview

- Java™ Implementation

- JavaScript implementation

- Examining the result

Overview

- Many enterprises today decide to develop their own customer or employee-facing mobile applications.

- Some of those companies already have mobile-facing websites (mobile web).

- An important decision must be made by companies:

  - Should all the mobile web features be implemented from scratch in the mobile application, which is great from a user experience perspective, but very time- and money-consuming?

  - Should the mobile application contain only new features and old ones should still be accessible by a mobile browser, which is easier to implement but not great in terms of user experience?
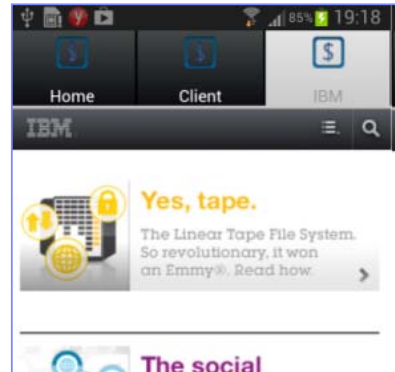
Overview

- The **webViewOverlay** approach allows the reuse and integration of existing mobile websites within a new mobile application.

- Navigation is smooth and seamless between components that are internal in the mobile application and the external content on the external mobile website.

Web resources bundled inside the application
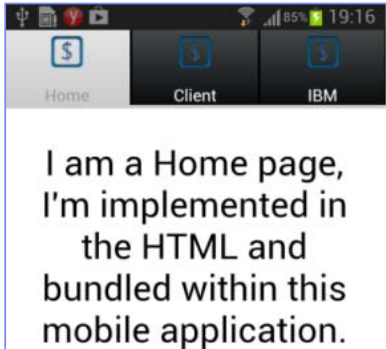
External web content

Overview

- In this module, you create a simple application with three tab bars.
- The first two tabs will contain internal content, the third bar will show an external IBM mobile website.

First and second tabs contain internal web resources

Third tab looks like another application page

But technically, it contains an additional WebView component on top of it

Overview

- **WebViewOverlay** is implemented by using the Apache Cordova plug-ins functionality.

- Developers can easily create their own protocol between internal web components and the **WebViewOverlay** control.

- You can use this module sample to understand the concepts of **WebViewOverlay.**

- Make sure that you are proficient with implementing Cordova plug-ins before proceeding.

Java implementation

- First, declare a **webViewOverlay** object to display the external content. Use static references for simplicity.

```java
public class WebViewOverlaySample extends WLDroidGap {

    private static WebView webViewOverlay;
    public static Activity thisapp;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl(getWebMainFilePath());

        thisapp = this;

        WebViewClient webViewClient = new WebViewClient() {

        webViewOverlay = new WebView(this);
        webViewOverlay.setVisibility(View.INVISIBLE);
        webViewOverlay.setWebViewClient(webViewClient);

        RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(
                RelativeLayout.LayoutParams.FILL_PARENT,
                RelativeLayout.LayoutParams.FILL_PARENT);
        webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);
        webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);
        webViewOverlay.getSettings().setJavaScriptEnabled(true);
```

Java implementation

- Create it, set its layout properties, and add it as a view to the root element. It is invisible initially.

```java
public class WebViewOverlaySample extends WLDroidGap {

    private static WebView webViewOverlay;
    public static Activity thisapp;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl(getWebMainFilePath());

        thisapp = this;

        WebViewClient webViewClient = new WebViewClient() {

        webViewOverlay = new WebView(this);
        webViewOverlay.setVisibility(View.INVISIBLE);
        webViewOverlay.setWebViewClient(webViewClient);

        RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(
                RelativeLayout.LayoutParams.FILL_PARENT,
                RelativeLayout.LayoutParams.FILL_PARENT);
        webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);
        webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);
        webViewOverlay.getSettings().setJavaScriptEnabled(true);
```

Java implementation

- Note the setMargins API. This API can be used to position your **webViewOverlay** component on a screen.

```java
public class WebViewOverlaySample extends WLDroidGap {

    private static WebView webViewOverlay;
    public static Activity thisapp;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl(getWebMainFilePath());

        thisapp = this;

        WebViewClient webViewClient = new WebViewClient() {

        webViewOverlay = new WebView(this);
        webViewOverlay.setVisibility(View.INVISIBLE);
        webViewOverlay.setWebViewClient(webViewClient);

        RelativeLayout.LayoutParams webViewOverlayLayoutParams = new RelativeLayout.LayoutParams(
                RelativeLayout.LayoutParams.FILL_PARENT,
                RelativeLayout.LayoutParams.FILL_PARENT);
        webViewOverlayLayoutParams.setMargins(0, 65, 0, 0);
        webViewOverlay.setLayoutParams(webViewOverlayLayoutParams);
        webViewOverlay.getSettings().setJavaScriptEnabled(true);
```

Java implementation

- Next, you need to set up your content view.

- Create a **RelativeLayout** object that will function as a root layout.

- Remove the **root** instance from its original parent.

- Add **root** and **webViewOverlay** to **rootRelativeLayout**.

- Set the content view to **rootRelativeLayout**.

```
    RelativeLayout rootRelativeLayout = new RelativeLayout(this);

    ((FrameLayout)root.getParent()).removeAllViews();
    rootRelativeLayout.addView(root);
    rootRelativeLayout.addView(webViewOverlay);
    setContentView(rootRelativeLayout);
}
```

Java implementation

- Implement the Java™ code of the Cordova plug-in.

```java
public class WebViewOverlayPlugin extends CordovaPlugin {

    private final String ACTION_OPEN_URL = "open";
    private final String ACTION_CLOSE_WEBVIEWOVERLAY = "close";

    @Override
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext) {

        if (action.equals(ACTION_OPEN_URL)) {
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {
                public void run() {
                    WebViewOverlaySample.clearWebViewOverlayHistory();
                    WebViewOverlaySample.loadWebViewOverlay("http://m.ibm.com/");
                    WebViewOverlaySample.setWebViewOverlayVisibility(View.VISIBLE);
                    WebViewOverlaySample.requestWebViewOverlayFocus();
                    WebViewOverlaySample.clearWebViewOverlayHistory();
                }
            });
```

ACTION_CLOSE_WEBVIEWOVERLAY)) {
thisapp.runOnUiThread(new Runnable() {
{
ySample.loadWebViewOverlayContent("");
ySample.setWebViewOverlayVisibility(View.INVISIBLE);
ySample.clearWebViewOverlayHistory();

Declare the actions that the plug-in will support.

```java
}
```

Java implementation

- Implement the Java code of Cordova plug-in.

```java
public class WebViewOverlayPlugin extends CordovaPlugin {

    private final String ACTION_OPEN_URL = "open";
    private final String ACTION_CLOSE_WEBVIEWOVERLAY = "close";

    @Override
    public boolean execute(String action, JSONArray args, CallbackContext callbackContext) {

        if (action.equals(ACTION_OPEN_URL)) {
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {
                public void run() {
                    WebViewOverlaySample.clearWebViewOverlayHistory();
                    WebViewOverlaySample.loadWebViewOverlay("http://m.ibm.com/");
                    WebViewOverlaySample.setWebViewOverlayVisibility(View.VISIBLE);
                    WebViewOverlaySample.requestWebViewOverlayFocus();
                    WebViewOverlaySample.clearWebViewOverlayHistory();
                }
            });
            return true;
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {
                                          .thisapp.runOnUiThread(new Runnable() {
                                           ) {
                                          aySample.loadWebViewOverlayContent("");
                                          aySample.setWebViewOverlayVisibility(View.INVISIBLE);
                                          aySample.clearWebViewOverlayHistory();
```

If an "open" request was received from the web part of the application, load the external content and make webViewOverlay visible.

Java implementation

- Implement the Java code of Cordova plug-in.

```java
public class WebViewOverlayPlugin extends CordovaPlugin {

                              EN_URL = "open";
                              OSE_WEBVIEWOVERLAY = "close";


                              action, JSONArray args, CallbackContext callbackContext) {

                              PEN_URL)) {
                              hisapp.runOnUiThread(new Runnable() {
                              {
                              Sample.clearWebViewOverlayHistory();
                              Sample.loadWebViewOverlay("http://m.ibm.com/");
                              Sample.setWebViewOverlayVisibility(View.VISIBLE);
                              Sample.requestWebViewOverlayFocus();
                              Sample.clearWebViewOverlayHistory();
                    }
                });
                return true;
        } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {
            WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {
                public void run() {
                    WebViewOverlaySample.loadWebViewOverlayContent("");
                    WebViewOverlaySample.setWebViewOverlayVisibility(View.INVISIBLE);
                    WebViewOverlaySample.clearWebViewOverlayHistory();
                }
            });
            return true;
        } else
            return false;
    }

}
```

After you receive a "close" request from the web part of the application, clean the webViewOverlay and hide it.

Java implementation

- Implement the Java code of Cordova plug-in.

```java
public class WebViewOverlayPlugin extends CordovaPlugin {

    private final String ACTION_OPEN_URL    "    "
    private final String

    @Override
    public boolean exec                                                     callbackContext) {

        if (action.equa
            WebViewOver
                public
                    Web
                    Web                                                     com/");
                    Web                                                     VISIBLE);
                    Web
                    Web
            }
        });
        return true;
    } else if (action.equals(ACTION_CLOSE_WEBVIEWOVERLAY)) {
        WebViewOverlaySample.thisapp.runOnUiThread(new Runnable() {
            public void run() {
                WebViewOverlaySample.loadWebViewOverlayContent("");
                WebViewOverlaySample.setWebViewOverlayVisibility(View.INVISIBLE);
                WebViewOverlaySample.clearWebViewOverlayHistory();
            }
        });
        return true;
    } else
        return false;
    }

}
```

Notice that the UI-related actions should be performed on a dedicated UI thread.

JavaScript implementation

- In JavaScript, create a **WebViewOverlayPlugin** object and populate it with the required methods.

```javascript
function WebViewOverlayPlugin() {};

WebViewOverlayPlugin.prototype.open = function(successCallback, errorCallback) {
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'open', []);
};

WebViewOverlayPlugin.prototype.close = function(successCallback, errorCallback) {
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'close', []);
};

WebViewOverlayPlugin.prototype.goBack = function(successCallback, errorCallback) {
    cordova.exec(successCallback, errorCallback, 'WebViewOverlayPlugin', 'goBack', []);
};

cordova.addConstructor(function() {
    if (!window.plugins) window.plugins = {};
    window.plugins.webViewOverlay = new WebViewOverlayPlugin();
});
```

JavaScript implementation

- Analyze the clicked tab ID and show or hide the webViewOverlay accordingly.

```javascript
function tabClicked(id){
    WL.Logger.debug("tabClicked >> id :: " + id);
    $(".tab").addClass('hidden');

    if (id=="3"){
        openOverlayWebView();
    } else {
        $("#tab" + id).removeClass('hidden');
        closeOverlayWebView();
    }
}

function openOverlayWebView(){
    WL.Logger.debug("openOverlayWebView");
    busyIndicator.show();
    setTimeout(busyIndicator.hide, 5000);
    window.plugins.webViewOverlay.open();
}

function closeOverlayWebView(){
    WL.Logger.debug("closeOverlayWebView");
    window.plugins.webViewOverlay.close();
}
```
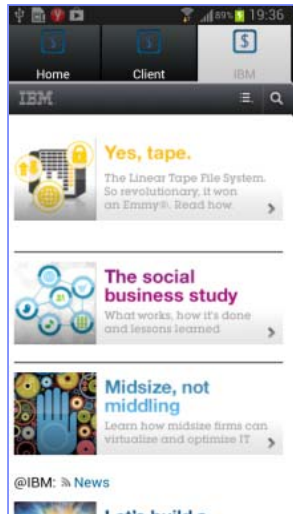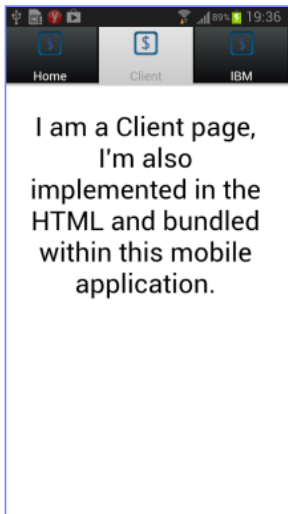
Loading external content can take time, so add a busyIndicator to improve the user experience.

Examining the result

- The sample for this training module can be found in the Getting Started page of the IBM Worklight® documentation website at http://www.ibm.com/mobile-docs.
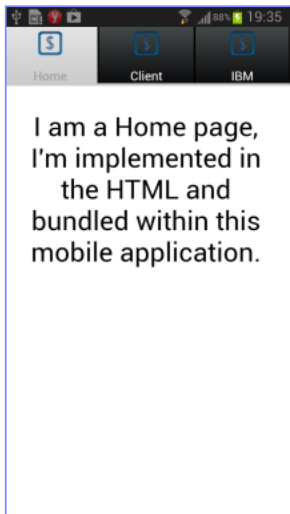
Notices

- Permission for the use of these publications is granted subject to these terms and conditions.

- This information was developed for products and services offered in the U.S.A.

- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

  – IBM Director of Licensing
    IBM Corporation
    North Castle Drive
    Armonk, NY 10504-1785
    U.S.A.

- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

  – Intellectual Property Licensing
    Legal and Intellectual Property Law
    IBM Japan Ltd.
    1623-14, Shimotsuruma, Yamato-shi
    Kanagawa 242-8502 Japan

- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

  – IBM Corporation
    Dept F6, Bldg 1
    294 Route 100
    Somers NY 10589-3216
    USA

- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

  – © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.
    © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.

- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
  - http://www.ibm.com/mobile-docs
- **Support**
  - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
    - http://www.ibm.com/software/passportadvantage
  - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
    - http://www.ibm.com/support/handbook
- **Comments**
  - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
  - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
  - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
  - Thank you for your support.
  - Submit your comments in the IBM Worklight Developer Edition support community at:
    - https://www.ibm.com/developerworks/mobile/worklight/connect.html
  - If you would like a response from IBM, please provide the following information:
    - Name
    - Address
    - Company or Organization
    - Phone No.
    - Email address

Thank You